Nagios XI - NRPE Troubleshooting and Common Solutions

Nagios®

The Industry Standard in IT Infrastructure Monitoring

Purpose

This document describes how to troubleshoot NRPE (Nagios Remote Plugin Executor) agent issues in Nagios® XI™. NRPE is most commonly used to monitor other Linux or Unix servers. This document will cover solutions to common problems/errors and general troubleshooting tips. Additionally, it provides a simple framework to systematically drill down problems with NRPE.

Target Audience

This document is intended for Nagios administrators who are having issues with the NRPE agent on Linux platforms. It assumes that you have some basic knowledge of how NRPE works, and that you have installed the NRPE agent on a remote server.

Prerequisites

There are a number of prerequisites and defaults that need to be explained before running through this document:

- This document will assume that NRPE and the plugins are installed in the default directories. If your corporate build or repo installed NRPE or the Nagios-plugins to a different directory than the default location, you will have to make note and substitute your paths for those in this document. The directories referenced most in this document are /usr/local/nagios/libexec and /usr/local/nagios/etc.
- The remote system being checked by the NRPE agent will be referred as “remote host” while the Nagios server will be referred to as the “XI Server” or “Nagios Server”.
- The command line editor “nano” will be used throughout this document. You are welcome to use vim/emacs/etc, but for simplicity's sake, “nano” will be referenced.
- Any packages that must be installed for certain sections of this document will use the standard Red Hat/CentOS package manager “yum”. If you are using a different distribution, you most likely will have to install the packages through your package manager (apt-get for Debian/Ubuntu, pacman for Archlinux, etc).

What Types Of Problems Can An Administrator Expect To Encounter With NRPE?

The most common problems are found in the initial setup of NRPE. Connection and communication issues are some of the easiest to troubleshoot and resolve. Other problems that people run into involve specific errors reported by Nagios XI in reference to an agent or remote host.

Error Codes And Other Issues Covered In This Document

This is by no means a definitive list, but these are the most common problems associated with the NRPE agent. Although it is not entirely in the scope of this document, there are a few tips for troubleshooting NSClient++ when using the NRPE handler.

Errors:

I. Return code of 127 is out of bounds - plugin may be missing
II. Return code of 126 is out of bounds - plugin may not be executable
III. CHECK_NRPE: Error - Could not complete SSL handshake
IV. CHECK_NRPE: Socket timeout after n seconds
V. CHECK_NRPE: Received 0 bytes from daemon. Check the remote server logs for error messages
VI. CHECK_NRPE: Error receiving data from daemon(SSL on but not used, too short timeout)
VII. NRPE: Unable to read output (path in nrpe.cfg wrong)
VIII. Command '[your plugin]' not defined

Copyright © 2014 Nagios Enterprises, LLC

Revision 1.0 – September, 2015

IX. Connection refused by host
X. No output returned from plugin
XI. Error while loading shared libraries: libssl.so.0.9.8: cannot open shared object file: No such file or directory
XII. Warning: This plugin must be either run as root or setuid
XIII. Connection refused or timed out

NSClient++ NRPE Specific Errors:

XIV. UNKNOWN: No handler for that command
XV. ERROR: Missing argument exception

XVI. General Troubleshooting Steps

If you are experiencing an error with NRPE that is not listed here, you are encouraged to contact us at the Nagios Support Forum for possible resolutions:

http://support.nagios.com/forum

I. Return Code Of 127 Is Out Of Bounds - Plugin May Be Missing

This error is usually experienced when the plugin referenced by the command directive in nrpe.cfg is either missing from the libexec folder or the command directive is named incorrectly. It could also imply that the command name passed through NRPE from the Nagios XI server is not defined in the nrpe.cfg file on the remote host. The first troubleshooting step is to make sure the plugin exists on the remote host. For this example will will be using the command check_foo.

Open a terminal and log into the remote host server as root, then execute the following command:

```
ls /usr/local/nagios/libexec
```

You should see the name of the plugin, in this example check_foo listed in the output. If not, you will have to copy the plugin to the /usr/local/nagios/libexec folder.

```
-rwxr-xr--. 1 root   nagios     2289 Nov 21 01:39 check_foo.sh
```

If the plugin file exists, check the nrpe.cfg file on the remote host.

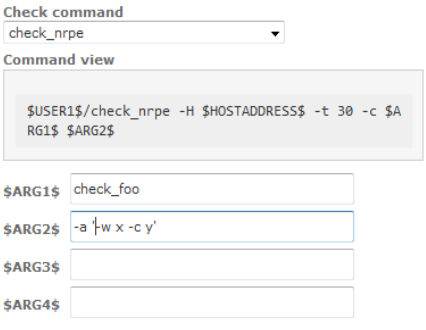
```
nano /usr/local/nagios/etc/nrpe.cfg
```

You will find the commands defined near the bottom of the file. Commands will be in the following format within the nrpe.cfg:

```
command[check_foo]=/usr/local/nagios/libexec/check_foo.sh $ARG1$
```

Verify a command declaration for the plugin exists and the path '/usr/local/nagios/libexec/check_foo.sh' matches the path of the plugin verified above. Some plugins have file extensions (.sh, .bin, .pl, .py, etc.). The path must include the extension of the plugin, but the command directive name, wrapped in 'command[]' does not need an extension.

Note: The command directive name, wrapped in 'command[]' , can be named something entirely different than the plugin file itself. This way you can use the same plugin for multiple command directives with different command names.



Next navigate to **Configure** → **Core Config Manager** → **Services** in the Nagios XI interface and select your service check.

The format for an NRPE check in Nagios XI is as follows:

1. Check command: check_nrpe
2. $ARG1$: the command name, for example: check_foo
3. $ARG2$: arguments to be passed to the plugin.

Copyright © 2014 Nagios Enterprises, LLC
Revision 1.0 – September, 2015

Verify the spelling of "*check_foo*" in $ARG1$ matches the exact spelling of the command directive name, "command[*check_foo*]" from the nrpe.cfg on the remote host.

II. Return Code Of 126 Is Out Of Bounds - Plugin May Not Be Executable

Many times when a plugin is downloaded from the exchange and copied to the remote host, it will not have executable permissions. You can verify this by getting a long-listing of the libexec plugin directory. For this example will will be using the command check_foo. Log into the remote host server as root and execute the following command:

```
ls -l /usr/local/nagios/libexec
```

You should see a listing similar to:

```
-rwxr-xr-x. 1 root   root     4173 Nov 21 01:39 check_bl
-rw-r--r--. 1 root   root     2289 Nov 21 01:39 check_foo.sh
```

The far left column of the listing are the permissions for each file. If you noticed, "check_foo.sh" is missing an "x" in a few places. These are executable permissions and can easily be added to the file using the following command:

```
chmod +x /usr/local/nagios/libexec/check_foo.sh
```

Remember that "check_foo.sh" is just an example and you will change /usr/local/nagios/libexec/check_foo.sh to the actual name and path to your plugin that is missing executable permissions.

III. CHECK_NRPE: Error - Could Not Complete SSL Handshake

Allowed hosts:
This is probably the most common of all error messages and one of the first you will experience when new to NRPE. There are a few different causes of this, though the most likely one is that the Nagios server's IP address is not defined in the remote host's nrpe.cfg file. Log into the remote host as the root user and edit the nrpe.cfg file:

```
nano /usr/local/nagios/etc/nrpe.cfg
```

You will need to add the IP address of your Nagios server is listed as an allowed host. Look for the line: `allowed_hosts=127.0.0.1` and change:
`allowed_hosts=127.0.0.1`

To:
`allowed_hosts=127.0.0.1,<nagios server ip>`

Remember to use your <actual nagios server IP address> and do not copy the above example verbatim. The allowed_hosts is a comma-separated list of IP addresses which can execute NRPE commands.

If you use xinetd for controlling the NRPE daemon (most people do), then you need to add the Nagios server's IP address to the xinetd NRPE configuration file: /etc/xinetd.d/nrpe.

```
nano /etc/xinetd.d/nrpe
```

In this file you will find the line: `only_from = 127.0.0.1` **This list is space-delimited list** (instead of comma delimited like the nrpe.cfg allowed_hosts directive). Change: `only_from = 127.0.0.1`

To: `only_from = 127.0.0.1 <Nagios server ip>`

Again, remember to use your actual nagios server IP address. One thing to note is that localhost (127.0.0.1) should remain as it allows you to troubleshoot NRPE issues locally. After you have made the following changes, restart the NRPE service on the remote host to bring up NRPE with the new configuration options.

Copyright © 2014 Nagios Enterprises, LLC
Revision 1.0 – September, 2015

If you use xinetd:
```
service xinetd restart
```

If you use an init-script method (this is the default way, but your distribution may vary):
```
/etc/init.d/nrpe restart
```

SSL Not Compiled In:
The other common cause is that NRPE was not compiled with ssl enabled. To recompile NRPE with ssl support, browse to your NRPE source directory (usually in /tmp/nrpe-2.14 if you followed the compiling NRPE from source document) and re-compile using the `–enable-ssl` flag:

```
cd /tmp/nrpe-2.14
./configure --enable-ssl
make all
make install
```

Understand that if you installed from a corporate build or from a package repo, you may have either uninstall the current NRPE package and install from source. You may need to pursue support on the specific distribution's forums or through Nagios support.

Xinetd Per Source Limit:
This cause is rare, but worth mentioning. If you use your remote host's NRPE server as a NRPE node proxy (sending all checks for the network segment to a single NRPE enabled server behind a firewall), or if you are doing a large number of NRPE checks in relatively short time period on one remote host, you may hit the maximum connection limit of NRPE. This is technically an xinetd setting and can be uncapped by editing the file /etc/xinetd.d/nrpe on your remote host:

```
nano /etc/xinetd.d/nrpe
```

Add the following line to the file inside the closing "}":

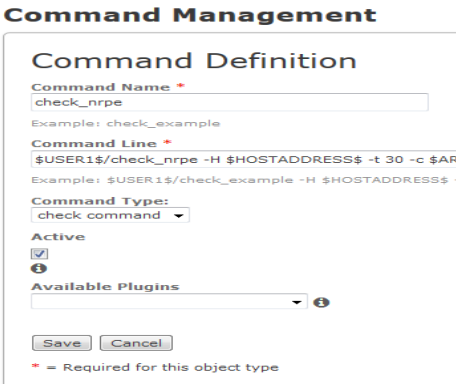
```
per_source = UNLIMITED
instances = UNLIMITED
```

And then restart NRPE with the following command:

```
service xinetd restart
```

IV. CHECK_NRPE: Socket Timeout After n Seconds

Increase Socket Timeout:
This is one of the harder to pin down errors. More often than not, following the steps from **part III** will be enough to solve this problem. But sometimes, it is not related to SSL or your allowed hosts. In these instances, it can either be that a plugin is taking longer than "n" seconds to return the check, or there is a firewall/port issue.



You can increase the timeout on the check, though you will have to alter the check in XI and the command and connection timeout in the nrpe.cfg file on the remote host. By default the timeout is set to 10 seconds, which is too short for certain checks (disk/filesystem/database checks among others). You can specify the timeout in XI by including the switch "-t" in the check_nrpe command.

In the Nagios XI web interface, go to **Configure → Core Config Manager → Commands**. This brings up the Commands page and you can enter *NRPE* into the **Search** field and click **Search**. Finally select the "check_nrpe" command.

In the **Command Line**, change "-t 10" to a higher value, we will use 30 seconds in this example ("-t 30"). **Save** your changes and then press the **Apply Configuration** button.

You may need to change a couple settings in the remote host's

Copyright © 2014 Nagios Enterprises, LLC

/usr/local/nagios/etc/nrpe.cfg file depending on how high you set the timeout in Nagios XI.

```
nano /usr/local/nagios/etc/nrpe.cfg
```

Search for the "command_timeout=" and "connection_timeout=" settings which may need to be altered. Set both of these, at minimum, to the value of the timeout in Nagios XI. Usually the "connection_timeout=300" is more than enough, as is the command_timeout which defaults to 60 seconds. If you do set your timeout in Nagios XI higher, increase the command_timeout to match.

Check the NRPE Service Status:
You may receive this error if the NRPE daemon is not running on the remote host. If you are using xinetd, you can check the status of the service by logging onto the remote host as root and running the following command:

```
service xinetd status
```

You should see output similar to the following:

```
       xinetd (pid  1260) is running...
```

If you are using the init-script method, or if your distribution does not use the "service" command, you can always grep a process listing:

```
ps -aef | grep nrpe
```

You should see output similar to the following (important bits in bold):

```
     nagios   53213     1  0 Feb26 ?    00:00:07 /usr/libexec/nrpe -c /etc/nagios/nrpe.cfg --daemon
```

If NRPE/xinetd is not running, start it with the following command:

```
service xinetd start
```

Or if you are not using xinetd:

```
/path/to/init/script start
```

Check Firewall and Port Settings:
The last of the probable causes of this error is associated with firewalls and ports. If the NRPE traffic is not traversing a firewall, you will see the checks timeout. Additionally, if port 5666 is not open on the remote host's firewall, you may receive a timeout error as well. Usually xinetd will open the ports automatically, as long as the /etc/xinetd.d/nrpe file is configured correctly, and NRPE's port settings have been added to /etc/services.

First, we should make sure that port 5666 is open on the remote host. The easiest way to do this, is to just run check_nrpe from the remote host to itself. This will also double as a good way to check that NRPE is functioning as expected. Log into the remote host as root and execute:

```
/usr/local/nagios/libexec/check_nrpe -H localhost
```

You should get something similar to the following output:

```
       NRPE v2.14
```

If not, make sure the that port 5666 is open on the remote host's firewall. If you are using xinetd go back to previous step (check the NRPE service status) as it should automatically open the port for you.

Checking Remote Host's Ports and Configuring Iptables:
This is usually for the init script method only. If you use an init script method, you may have to open port 5666 on your firewall, which in the case of most Linux distributions, is iptables. To get a listing of the current iptables rules, run the following on the remote host as root:

```
iptables -L
```


Copyright © 2014 Nagios Enterprises, LLC
Revision 1.0 – September, 2015

The expected output is similar to:

```
ACCEPT - tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:5666
```

If the port is not open, you will have to add an iptables rule for it.

```
nano /etc/sysconfig/iptables
```

Add the line:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 5666 -j ACCEPT
```

Save the file and restart NRPE:

```
/path/to/init/script restart
```

Checking Port 5666 From the Nagios XI Server with Nmap or Telnet:
You can use telnet or nmap (among other port scanners) to check the remote host's ports. If you do not have either of those packages, install one of them with yum for RHEL/CentOS systems:

```
yum install nmap
```

Or:

```
yum install telnet
```

Once installed, test the connection on port 5666 from the Nagios XI server to the remote host by logging in as root on your nagios server and running the following command:

```
nmap <remote host ip> -p 5666
```

Remember to replace your remote host server ip address above. The expected output should be similar to:

```
PORT     STATE SERVICE
5666/tcp open  nrpe
```

Alternatively, test with telnet:

```
telnet <remote host ip> 5666
```

Remember to replace your remote host server ip address above. The expected output should be similar to:

```
Trying <remote host ip>...
Connected to <remote host ip>.
```

V. CHECK_NRPE: Received 0 Bytes From Daemon. Check The Remote Server Logs For Error Messages

First, make sure that NRPE is running as this is a common cause of this error. For instructions on how to do so, refer to section IV of this document under **Check the NRPE Service Status**.

The other causes all deal with arguments. If you are passing arguments to the remote host through NRPE, the argument usage should be consistent between the Nagios XI service check and the arguments declared in the command directive in the remote host's nrpe.cfg. Additionally, check the remote host's nrpe.cfg for the “dont_blame_nrpe” directive. Log into the remote host as the root user and execute:

```
cat /usr/local/nagios/etc/nrpe.cfg | grep blame
```
The expected output should be:

Copyright © 2014 Nagios Enterprises, LLC

```
dont_blame_nrpe=1
```

Without this directive set to “1”, arguments will not be accepted for any checks other than those specified in the nrpe.cfg file itself. If the “dont_blame_nrpe” directive is set to “0”, you will need to edit /usr/local/nagios/etc/nrpe.cfg and set dont_blame_nrpe=1.

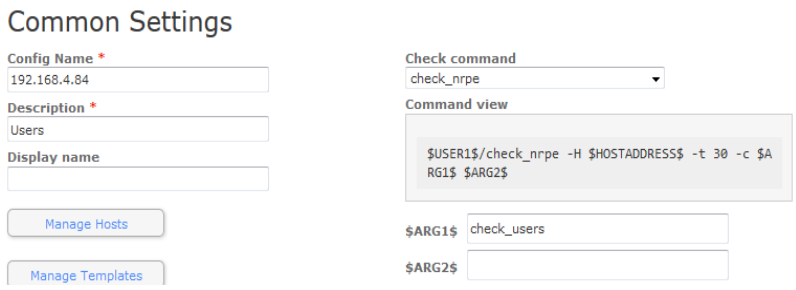
No Arguments
To verify if your argument usage is consistent, compare the check in Nagios XI to the command directive in the remote host's /usr/local/nagios/etc/nrpe.cfg file. If you have declared all the arguments for a check in the nrpe.cfg file, then Nagios XI should pass no arguments other than the command itself. In the example below, the command directive check_users is defined to not pass any arguments:

```
command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
```

We can verify the the arguments which are sent to the remote host from Nagios XI by navigating to **Configure → Core Config Manager → Services** and select the check_user service for your remote host.

As you can see the service check is created to send no arguments other than the command name in $ARG1$:

```
check_command: check_nrpe
$ARG1$ check_users
$ARG2$+ <blank>
```



Separate Arguments
If you have setup multiple arguments for each threshold/option, Nagios should pass them in the same order:

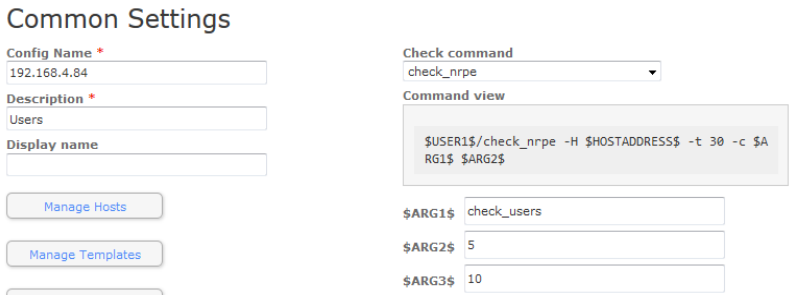
nrpe.cfg command directive:

```
command[check_users]=/usr/local/nagios/libexec/check_users -w $ARG1$ -c $ARG2$
```

The service check is set up in Nagios XI:

```
check_command: check_nrpe
$ARG1$ check_users
$ARG2$ 5
$ARG3$ 10
$ARG4$+ <blank>
```

Notice the command directive expects $ARG1$ and $ARG2$ even though in Nagios they are actually $ARG2$ and $ARG3$. This trips up beginners, as Nagios passes all 3 arguments to check_nrpe, where check_nrpe then passes the command and it's 2 arguments to NRPE on the remote host. Just remember that in check_nrpe uses the first argument to pass the command name, all other arguments are specific to the command you are executing on the remote host.



Combined Arguments
The final format is to encapsulate all of the arguments into one field in Nagios XI and one $ARG1$ in the remote host's nrpe.cfg file. This is how Nagios sets up checks configured through the linux-server and NRPE wizards, so if you compiled NRPE from source for the remote host but are using the XI wizards to create checks, you will have to edit the command directive in the remote host's nrpe.cfg file.

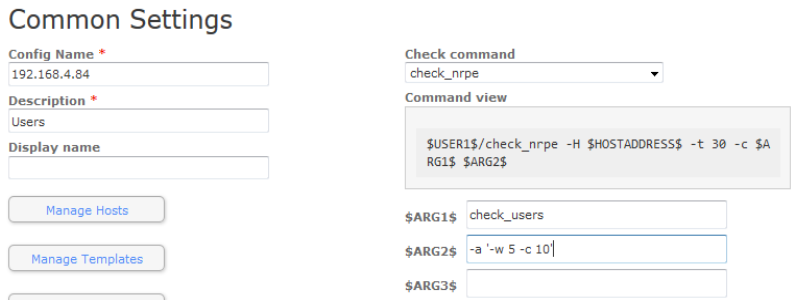
nrpe.cfg command directive:

Copyright © 2014 Nagios Enterprises, LLC
Revision 1.0 – September, 2015

```
command[check_users]=/usr/local/nagios/libexec/check_users $ARG1$
```

The service check is set up in Nagios XI:

```
check_command: check_nrpe
$ARG1$ check_users
$ARG2$ -a '-w 5 -c 10'
$ARG3$+ <blank>
```



All three of these argument configuration methods are valid, though it is best to choose one method and stick to it for consistency and ease of troubleshooting.

VI. CHECK_NRPE: Error Receiving Data From Daemon

This error is not to be confused with the error "CHECK_NRPE: Received 0 bytes from daemon" as they have separate causes. Most often, this error is experienced when passing the no ssl switch (-n) to check_nrpe even though NRPE on the remote host was compiled with ssl enabled. There are very few instances where NRPE is best run without ssl, so if you added the "-n" switch to your check for testing reasons, make sure to remove the switch before deploying the check. If you have a reason for not using ssl, do note that you will have to compile NRPE without ssl to avoid this error when using the "-n" switch.

The other general cause of this error, though rare, happens when your check's check_nrpe timeout is set too low. To increase the timeout, refer to section of this document named **IV. CHECK_NRPE: Socket Timeout After n Seconds** under the subsection **Increase Socket Timeout**.

VII. NRPE: Unable To Read Output

This error implies that NRPE did not return any character output. Common causes are incorrect plugin paths in the nrpe.cfg file or that the remote host does not have NRPE installed. Rarely, it is caused by trying to run a plugin that requires root privileges.

Incorrect Plugin Paths
First, log onto the remote host as root and check the plugin paths in /usr/local/nagios/etc/nrpe.cfg. Try to browse to the plugin folder and make sure the plugins are listed. Sometimes when installing from a package repo, the commands in nrpe.cfg will have a path to a distribution specific location. If the nagios-plugins package was installed from source or moved over from another remote host, they me be located in a different directory.

The default location for the nagios-plugins can be found at /usr/local/nagios/libexec. Open up your nrpe.cfg file on the remote host and take note of the path for the command directives (in bold):

```
command[check_users]=/usr/local/nagios/libexec/check_users $ARG1$
```

Change directory to this location and get a listing of this directories contents– you should see a large list of available plugins:

```
cd /usr/local/nagios/libexec/
ls
```

If the directory is blank or altogether missing, you are either missing the nagios-plugins, or they are in a different directory. You will need to change your nrpe.cfg file to reflect the location of your plugins.

Is NRPE Installed?
Next, make sure that NRPE is indeed installed on the remote host. Log onto the remote host as root and execute the following command:

```
find / -name nrpe
```

The results should be similar to the following:

Copyright © 2014 Nagios Enterprises, LLC
Revision 1.0 – September, 2015

```
/usr/local/nagios/bin/nrpe
/usr/local/nagios/etc/nrpe
---- Truncated --------
```

If NRPE is installed, refer to part IV of this document **CHECK_NRPE: Socket Timeout After n Seconds**, under the section **Check The NRPE Service Status** to make sure that NRPE is actually running.

If the remote host does not have NRPE, you will have to install it. This can be done in a few different ways. We suggest installing NRPE via the Linux agent provided by Nagios XI. Please reference the below link for instructions:

Installing the Linux NRPE Monitoring Agent:
http://assets.nagios.com/downloads/nagiosxi/docs/Installing_The_XI_Linux_Agent.pdf

However if you need to compile NRPE from source, please reference the link below for instructions:

Installing and Configuring NRPE from Source:
http://assets.nagios.com/downloads/nagiosxi/docs/Source_Based_NRPE_Installation_and_XI.pdf

The Plugin Requires "sudo" Privileges
Finally, it may be that your specific plugin requires root access. Depending on the Linux distribution on the remote host, you may have to consult the specific distribution's forums for instructions on how to give permission to the plugin and the user "nagios". For this example, we will use sudo and the /etc/sudoers file.

You will need to create a rule in /etc/sudoers for the user nagios and the plugin script/binary requiring root access. Additionally, if the plugin script calls another system binary that requires root access, you will need to specify a rule for that binary as well (this problem is most often found with raid array plugins that require an access to a third party utility that requires root access). Log into the remote host as root and edit the sudoers file:

```
nano /etc/sudoers
```

You will need to add the following line (replace <plugin> with the file name of your plugin):

```
nagios   ALL = NOPASSWD:/usr/local/nagios/libexec/<plugin>
```

If your plugin requires another binary on the system that is restricted to root, you will have to create an additional rule (replace `/path/to/binary` with the actual path to the required binary):

```
nagios   ALL = NOPASSWD:/path/to/binary
```

This will allow the user "nagios" (the user that NRPE runs as) to run the specified plugin as root (through sudo) without a password. You should be very careful with these settings, as incorrectly configuring it will lead to LARGE security vulnerabilities.

The final step is to add "sudo" to the command in the remote host's nrpe.cfg:

```
command[check_raid]=sudo /usr/local/nagios/libexec/check_raid
```

Now restart NRPE and verify the plugin is working correctly.

VIII. Command '[Your Plugin]' Not Defined

This error is very straight forward. Usually this is caused by a mismatch between the command name declared in Nagios XI to be check through NRPE and the actual command name of the command directive in the remote host's nrpe.cfg file. For more information see section **I. Return Code Of 127 Is Out Of Bounds - Plugin May Be Missing**.

IX. Connection Refused By Host

Copyright © 2014 Nagios Enterprises, LLC
Revision 1.0 – September, 2015

This error usually relates to port/firewall issues or improperly configured “allowed_hosts” directives. See the following sections of this document for the pertinent troubleshooting steps:

III. CHECK_NRPE: Error - Could Not Complete SSL Handshake
IV. CHECK_NRPE: Socket Timeout After n Seconds

X. No Output Returned From Plugin

There are a few causes of this error, two of which have solutions that have been covered other places in this document.

Permissions
The most common solution is to check the permissions on the check_nrpe binary on the Nagios XI server:

```
ls -la /usr/local/nagios/libexec/check_nrpe
```

The expected permissions should resemble:

```
-rwxrwxr-x. 1 nagios nagios  75444 Nov 21 01:38 check_nrpe
```

If not, change ownership to user/group “nagios” and fix up the permissions:

```
chown nagios:nagios /usr/local/nagios/libexec/check_nrpe
chmod u+rwx /usr/local/nagios/libexec/check_nrpe
chmod u+rx /usr/local/nagios/libexec/check_nrpe
```

This should be setup by default during the install process, but enough people have had the issues that it was worth noting here.

Missing Plugin
Another cause is a missing plugin file, though, in order to receive this error, you usually have to also be experiencing a secondary configuration issue. In order to resolve issues relating to missing plugins, see the section I. Return Code of 127 Is Out Of Bounds - Plugin May Be Missing for possible solutions.

Mismatch of Arguments between Nagios XI and nrpe.cfg
The final cause, and usually the secondary issue for those who found their plugin missing from the expected location, is an argument usage mismatch between the remote host's nrpe.cfg command directive and the arguments passed by Nagios through check_nrpe. This was covered in this document under the section V. CHECK_NRPE: Received 0 Bytes From Daemon.

XI. Error While Loading Shared Libraries: libssl.so.0.9.8: Cannot Open Shared Object File: No Such File Or Directory

You are probably missing the ssl libraries on the remote host. This is an easy fix, as all you need to do is install openssl from the host's distribution repos. For example, in CentOS/RHEL, log onto your remote host and execute the following command:

```
yum install openssl
```

You can verify that it installed correctly with:

```
which openssl
```

The output should be similar to:

```
/usr/bin/openssl
```

If you use another distribution other than CentOS or RHEL, you may need to consult with their forums or run a search with the distribution's package manager to locate the correct package.

Copyright © 2014 Nagios Enterprises, LLC
Revision 1.0 – September, 2015

XII. Warning: This Plugin Must Be Either Run As Root Or Setuid

This error is usually plugin specific and is most commonly experienced when trying to use a third-party hardware check plugin (most often disk smart checks and raid health plugins). You need to setup the sudoers file and associated config changes mentioned in this document earlier in the section **VII. NRPE: Unable To Read Output The Plugin**, subsection: **Requires 'sudo' Privileges**.

Sticky Bit
Alternatively, you could set the sticky bit on the plugin's permissions. Sudoers is considered safer, so only use this option if you understand the consequences:

```
chmod u+s /usr/local/nagios/libexec/<plugin>
```

XIII. Connection Refused Or Timed Out

This error is most often experienced when using the remote host as an NRPE proxy server to a network segment. It can also be caused by using an incorrect IP address or hostname in the check_nrpe command. (rare in Nagios XI configurations)
If you do use the remote host as an NRPE proxy, you may need to increase the maximum number of concurrent connections through xinetd. You need to add per_source = UNLIMITED to /etc/xinetd.d/nrpe. Log onto your remote host at root and execute:

```
nano /etc/xinetd.d/nrpe
```

Add the following line to the file inside the closing "}":

```
per_source=UNLIMITED
```

Restart xinetd:

```
service xinetd restart
```

NSClient++ NRPE Specific Errors:

XIV. UNKNOWN: No Handler For That Command

This is usually caused by a missing or incorrectly spelled handler (external alias) in the remote host's nsc.ini (v0.3.x) or nsclient.ini (v0.4.x). This file is typically found in c:\Program Files\NSClient++. Check the spelling of the check_nrpe command for the service check in Nagios XI (the name of the command after the "-c"). It should match the spelling of the external alias in the nsclient config file. For example:

```
[External Alias]
alias_cpu=checkCPU warn=80 crit=90 time=5m time=1m time=30s
...[truncated]...
```

In the example above, the bolded "alias_cpu" is the handler and therefore the service check in Nagios should specify the check_nrpe command as "alias_cpu".

XV. ERROR: Missing Argument Exception

This is usually due to clashing handler names (more than 1 of the same external alias name). It can also be caused by an argument mismatch as well. Read over the section **V. CHECK_NRPE: Received 0 Bytes From Daemon** of this document, specifically the **No Arguments** section for an in depth explanation of this problem.

Instead of editing the command directives in your nrpe.cfg file (which does not exist as this is a windows remote host), edit the "[External Alias]" section of C:\Program Files\NSClient++\NSC.ini (v0.3.x) or nsclient.ini (v0.4.x). Make sure your argument usage is consistent between the NSC.ini/NSClient,ini and the Nagios XI service check.

XVI. General Troubleshooting Tips

Copyright © 2014 Nagios Enterprises, LLC
Revision 1.0 – September, 2015

When Troubleshooting NRPE issues, there is a general order of procedures for drilling down the problem. Start with the plugin itself, and then move to NRPE, and finally check your argument usage. If you follow the general steps below before dealing with support, your issue may be solved faster than expected as these are always the first steps a Nagios XI support representative will ask you to perform:

1. Test The Plugin Locally First. Log onto your remote server as root and copy the plugin to your plugins directory (/usr/local/nagios/libexec) on the remote host and run it:

```
/usr/local/nagios/libexec/<name of plugin>
```

If it does not work as expected, you may want to check the plugin's usage as you may find some hints to why it is not working:

```
/usr/local/nagios/libexec/<name of plugin> -h
```

You may have to set some thresholds, usually warning (-w) and critical (-c) for a large number of plugins before they will work correctly. Once the plugin has been tested and working locally from the remote host, create a command directive for it in the nrpe.cfg file. Take a mental note of how you setup your arguments.

2. Verify That NRPE Is Working Locally And Open To Requests From The XI Server:

On the remote host, run:

```
service xinetd status
```

Or (for init script systems):

```
service nrpe status
```

If NRPE is not running, follow the steps in **Part III** of this document. If NRPE is running, move on to testing the connection to the remote host from the XI server with check_nrpe. Log onto the Nagios XI server as root and run the following command inserting the actual remote host IP address:

```
/usr/local/nagios/libexec/check_nrpe -H <remote host ip>
```

The command above should return the NRPE version of the remote host. If not, follow the steps in **Part IV** of this document. If the version of NRPE is returned successfully, move on to step 3.

3. Try The Full Command From The Command Line Interface On The XI Server:

From the Nagios XI command line interface, run the following command:

```
/usr/local/nagios/libexec/check_nrpe -H <remote host ip> -c <command and arguments>
```

You will need to replace the remote host IP address and match your command and arguments to your command directives in your remote host nrpe.cfg.

If you do not get the expected output, check the plugin usage again to make sure your syntax is correct. Refer to **Part VIII** of this document for information on argument usage. If the plugin does output the expected data, move on to step 4.

4. Setup The Service Check In XI:

Create a new service for the check by navigating within the Nagios XI web interface **Configure → Core Config Manager → Services → Add New.** Specify the **Config Name** and **Description** for the check. Use *check_nrpe* in the **Check_command** drop-down.

Next set up the command arguments under **Command view**.

$ARG1$ is the remote command to be sent to the remote host through NRPE. This must match the command directive in the nrpe.cfg.

$ARG2$ is used for extra command arguments. Again, if you have defined any in the remote host's nrpe.cfg..

Copyright © 2014 Nagios Enterprises, LLC
Revision 1.0 – September, 2015

The check needs to be applied to a host, so click the **Manage Hosts** button. Select a host from the list and click **Add Selected**. You should see the host appear in the right hand pane under **Assigned**. Now click **Close**.

Click the **Check Settings** tab. At minimum, we need to setup check intervals, attempts, and a period. **Check interval** specifies how often the check is run. **Retry interval** specifies the time between check retries when the service check has failed (SOFT STATE). **Max check attempts** specifies the number of retries a check will attempt before it is marked as a HARD STATE fail. The last required setting to set on this tab is the **Check period**. This specifies what "time period" the check should run and can be configured for certain days and time frames. **xi_timeperiod_24x7** will be fine for this example.

Last, click the **Alert Settings** and set the **Notification period** to "xi_timeperiod_24x7", or to the time period of your choice. This specifies the time period for notifications. (emails, SMS, etc.) Click **Manage Contacts** and add a contact to the check if you want. Finally, click **Save** and **Apply Configuration**.

Now when you navigate to **Service Detail** you will see your service check listed. It may take a minute for the service to change from pending to a STATE. From this page you can verify that your plugin is executing as expected.

Final Thoughts

Hopefully you have been able to resolve your issues related to NRPE using the information within this document. If you have any additional support related questions please visit us at our Nagios Support Forums:

http://support.nagios.com/forum

Copyright © 2014 Nagios Enterprises, LLC
Revision 1.0 – September, 2015